**A6-Write a program to analyze following packet formats captured through Wireshark for wired network. 1. Ethernet 2. IP 3.TCP 4. UDP**

#include <iostream>

#include<fstream>

#include <iomanip>

#include<string>

using namespace std;

int main() {

cout << "\*\*\*\*\* PACKET ANALYZER \*\*\*\*\*" << endl;

string value, sr\_no,time,source,destination,info,protocol,len;

int count=-1,i=0;

int choice;

do

{

ifstream file("a7.csv");

//Reinitialize Counters

count=-1;

i=0;

cout<<"\nEnter which protocol packets you want to see"<<endl;

cout<<"1.IP\n2.UDP\n3.TCP\n4.ARP\n5.Ethernet\n0Exit!!!\nChoice:"<<endl;

cin>>choice;

string protocolChoice; //sting to hold user packet choice

switch(choice){

case 1: protocolChoice="ICMPv6";

break;

case 2: protocolChoice="UDP";

break;

case 3: protocolChoice="TCP";

break;

case 4: protocolChoice="Ethernet";

break;

default: return 0;

break;

}

if(choice==5){

while(file.good()) //LOOP UNTIL FILE HAS CONTENT

{

getline(file,sr\_no,','); //GET STRING TILL ,

getline(file,time,',');

getline(file,source,',');

getline(file,destination,',');

getline(file,protocol,',');

getline(file,len,',');

getline(file,info,'\n');

protocol=string(protocol,1,protocol.length()-2);

cout <<setw(4)<<left<<i++;

cout <<setw(12)<<left<< string( time, 1, time.length()-2 );

cout << setw(30)<<left<<string( source, 1, source.length()-2 );

cout << setw(30)<<left<<string( destination, 1, destination.length()-2 );

cout <<setw(8)<<left<<protocol;

cout <<setw(8)<<left<< string( len, 1, len.length()-2 );

cout << string( info, 1, info.length()-2 )<<"\n";

count++;

}

goto end;

}

while(file.good()) //LOOP UNTIL FILE HAS CONTENT

{

getline(file,sr\_no,','); //GET STRING TILL ,

getline(file,time,',');

getline(file,source,',');

getline(file,destination,',');

getline(file,protocol,',');

getline(file,len,',');

getline(file,info,'\n');

protocol=string(protocol,1,protocol.length()-2);

if(protocol=="Protocol"||protocol==protocolChoice)

{

cout <<setw(4)<<left<<i++;

cout <<setw(12)<<left<< string( time, 1, time.length()-2 );

cout << setw(30)<<left<<string( source, 1, source.length()-2 );

cout << setw(30)<<left<<string( destination, 1, destination.length()-2 );

cout <<setw(8)<<left<<protocol;

cout <<setw(8)<<left<< string( len, 1, len.length()-2 );

cout << string( info, 1, info.length()-2 )<<"\n";

count++;

}

}

end:file.close();

cout<<"\nTotal Packet Count: "<<count;

}while(choice!=0);

return 0;

}

OUTPUT:

root@E6320:/home/nikhil# g++ assign7.cpp

root@E6320:/home/nikhil# ./a.out

\*\*\*\*\* PACKET ANALYZER \*\*\*\*\*

Enter which protocol packets you want to see

1.IP

2.UDP

3.TCP

4.ARP

5.Ethernet

0Exit!!!

Choice:

1

0 Time Source Destination ProtocolLength Info

ICMPv6 86 Multicast Listener Report

632 9.351792000 fe80::fabc:12ff:fe6b:6ac9 ff02::1:ff6b:6ac9 ICMPv6 86 Multicast Listener Report

633 9.370743000 fe80::9579:d8e5:90f4:1076 ff02::1:fff4:1076 ICMPv6 86 Multicast Listener Report

634 9.469462000 fe80::1a0f:76ff:fe8d:1fb6 ff02::1:ff62:f148 ICMPv6 86 Neighbor Solicitation for fe80::7876:c84c:e462:f148 from 18:0f:76:8d:1f:b6

635 9.603778000 fe80::ca1f:66ff:fe05:e8a9 ff02::1:ff05:e8a9 ICMPv6 86 Multicast Listener Report

636 9.608631000 fe80::fabc:12ff:fe6b:e9b5 ff02::1:ff6b:e9b5 ICMPv6 86 Multicast Listener Report

637 9.616955000 fe80::84eb:1f3c:4c35:64bc ff02::1:ff35:64bc ICMPv6 86 Multicast Listener Report

638 9.676905000 fe80::b1fe:4b23:7794:bec1 ff02::1:ff94:bec1 ICMPv6 86 Multicast Listener Report

639 9.823325000 fe80::995e:d9a:b8ac:cce ff02::1:ffac:cce ICMPv6 86 Multicast Listener Report

640 9.869314000 fe80::9824:6f1d:4757:6445 ff02::1:ff57:6445 ICMPv6 86 Multicast Listener Report

641 9.911430000 fe80::965f:4278:f3f9:2503 ff02::1:fff9:2503 ICMPv6 86 Multicast Listener Report

642 9.933923000 fe80::726d:ecff:fe02:230 ff02::1:ff00:0 ICMPv6 86 Multicast Listener Report

643 9.968563000 fe80::1a0f:76ff:fe8d:1fb6 ff02::1:ff62:f148 ICMPv6 86 Neighbor Solicitation for fe80::7876:c84c:e462:f148 from 18:0f:76:8d:1f:b6

644 9.996416000 fe80::c1a8:abbf:6c23:43b9 ff02::1:ff23:43b9 ICMPv6 86 Multicast Listener Report

645 9.999022000 fe80::28e0:ccf0:cf13:caa5 ff02::1:ff13:caa5 ICMPv6 86 Multicast Listener Report

646 10.041433000fe80::d8c0:4f49:d54f:7e3c ff02::1:ff4f:7e3c ICMPv6 86 Multicast Listener Report

647 10.041463000fe80::1a0f:76ff:fe8d:4370 ff02::1:ff40:9af5 ICMPv6 86 Neighbor Solicitation for fe80::daee:8335:9e40:9af5 from 18:0f:76:8d:43:70

648 10.055129000fe80::726d:ecff:fe02:19c ff02::1:ff00:0 ICMPv6 86 Multicast Listener Report

649 10.109614000fe80::fabc:12ff:fe82:75a ff02::1:ff82:75a ICMPv6 86 Multicast Listener Report

650 10.173865000fe80::fabc:12ff:fe6b:638e ff02::1:ff6b:638e ICMPv6 86 Multicast Listener Report

651 10.208296000fe80::bcdd:c9f1:eda7:ea86 ff02::1:ffa7:ea86 ICMPv6 86 Multicast Listener Report

Total Packet Count: 682

Enter which protocol packets you want to see

1.IP

2.UDP

3.TCP

4.ARP

5.Ethernet

0Exit!!!

Choice:

2

0 Time Source Destination ProtocolLength Info

1 2.820877000 10.10.11.43 255.255.255.255 UDP 82 Source port: 58462 Destination port: sentinelsrm

2 6.825433000 10.10.11.43 10.10.15.255 UDP 82 Source port: 58462 Destination port: sentinelsrm

Total Packet Count: 2

Enter which protocol packets you want to see

1.IP

2.UDP

3.TCP

4.ARP

5.Ethernet

0Exit!!!

Choice:

3

0 Time Source Destination ProtocolLength Info

1 9.507132000 216.58.220.5 10.10.15.30 TCP 66 https > 60112 [ACK] Seq=1 Ack=1112 Win=667 Len=0 TSval=494401162 TSecr=1554650

2 9.508036000 216.58.220.5 10.10.15.30 TCP 66 https > 60112 [ACK] Seq=1 Ack=1515 Win=678 Len=0 TSval=494401163 TSecr=1554651

3 9.751812000 10.10.15.30 216.58.220.5 TCP 66 60112 > https [ACK] Seq=1515 Ack=651 Win=1444 Len=0 TSval=1554902 TSecr=494401406

4 9.757968000 216.58.220.5 10.10.15.30 TCP 66 https > 60112 [ACK] Seq=651 Ack=1561 Win=678 Len=0 TSval=494401413 TSecr=1554902

Total Packet Count: 4

Enter which protocol packets you want to see

1.IP

2.UDP

3.TCP

4.Ethernet

0Exit!!!

Choice:

3

Enter which protocol packets you want to see

1.IP

2.UDP

3.TCP

4.Ethernet

0Exit!!!

Choice:

4

0 Time Source Destination ProtocolLength Info

Total Packet Count: 0